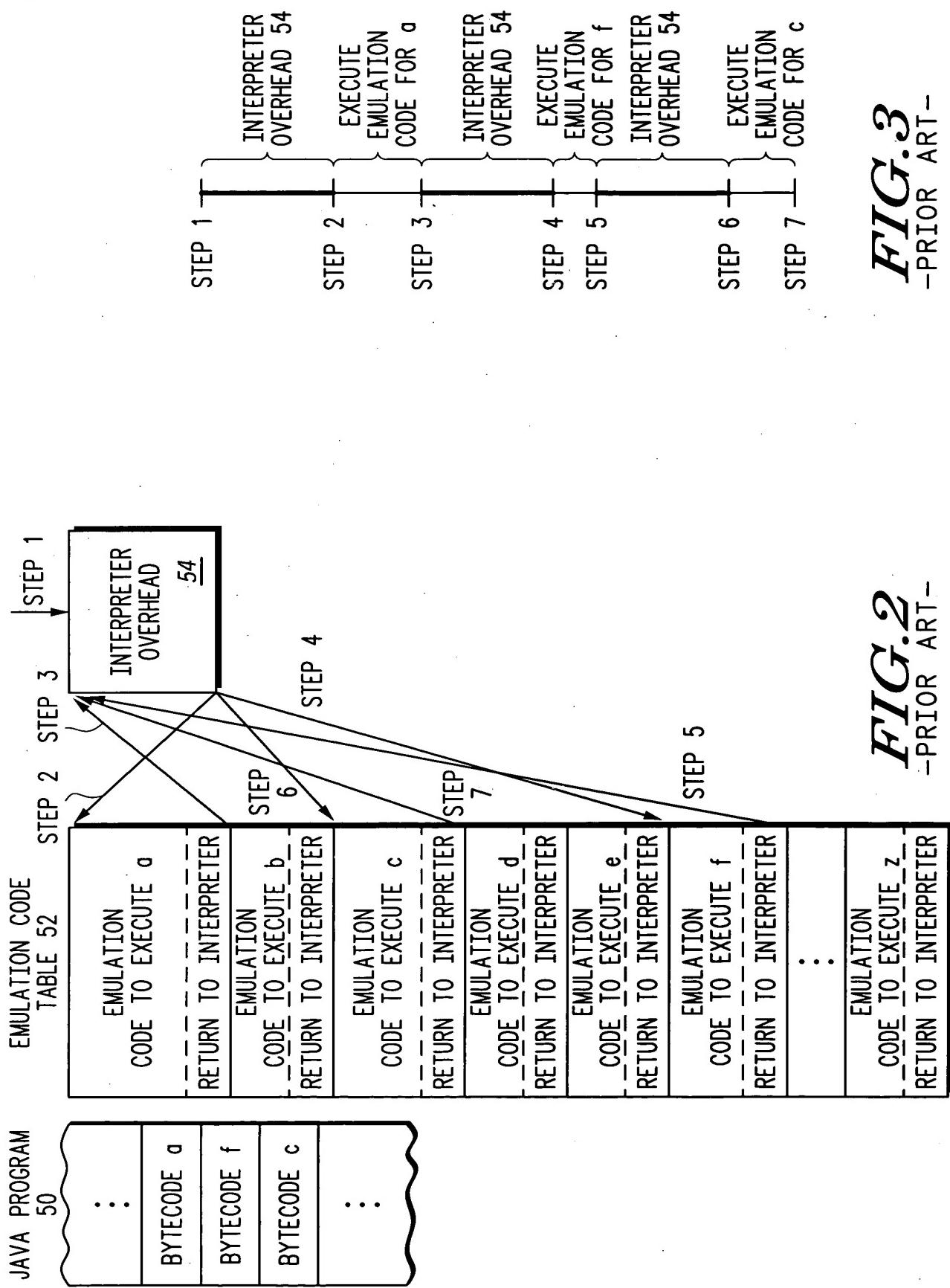
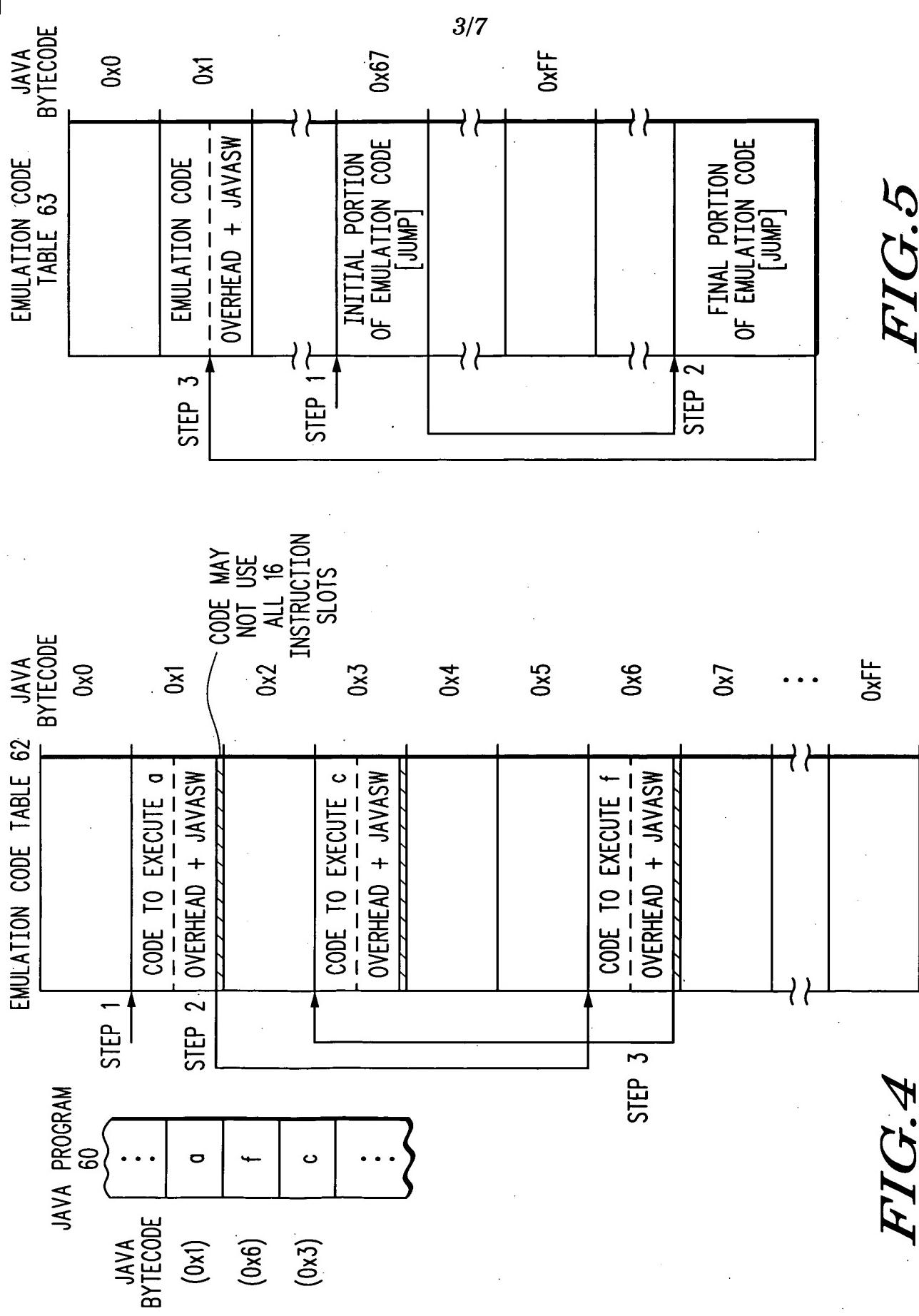


FIG.1





4/7

EMULATION CODE
TABLE 72

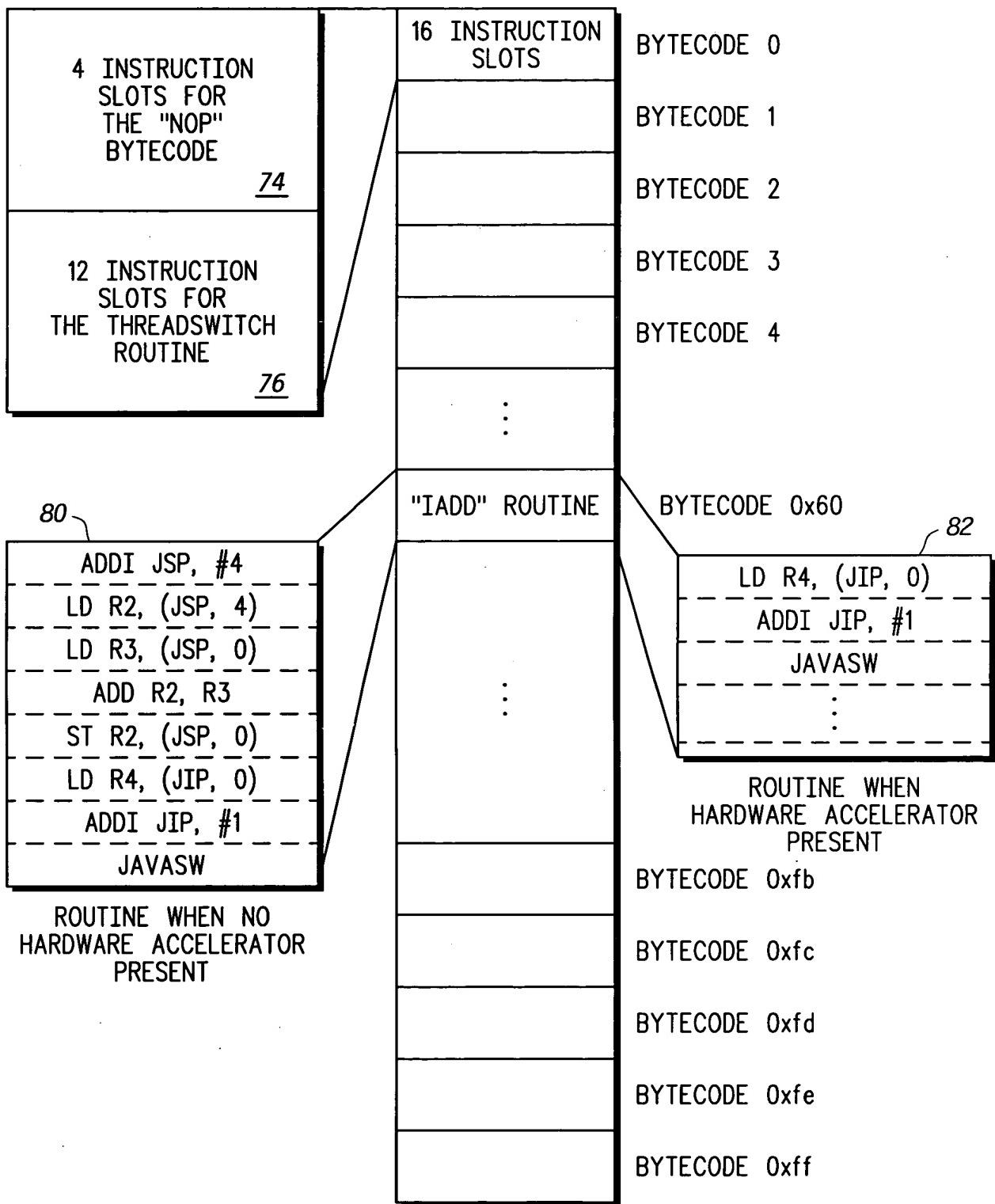


FIG. 6

JAVASW

JAVA INTERPRETER SWITCH

OPERATION: JUMP TO BYTECODE EMULATION SEQUENCE;

```
91 — IF (SWCOUNT!<0)
92 — PC<-[((PC+2) & 0xffffe000) | (R4<<5)];
93 — ELSE
94 — PC<-[((PC+2) & 0xffffe000) | (0x8)];
95 — SWCOUNT--;
```

ASSEMBLER

SYNTAX: JAVASW RX

DESCRIPTION: JUMP TO BYTECODE EMULATION SEQUENCE. THE LOW ORDER 13 BITS OF THE VALUE OF PC+2 ARE FORCED TO ZERO, AND BASED ON THE STATE OF THE SWCOUNT (IN REGISTER R12), EITHER A SCALED VALUE IN REGISTER RX IS LOGICALLY "OR'ED", OR A CONSTANT VALUE 0x8 OR'ED, AND INSTRUCTION EXECUTION RESUMES AT THE NEW PC VALUE. THE SWCOUNT REGISTER R12 IS DECREMENTED. NOTE THAT BECAUSE PC+2 IS USED AS THE BASE VALUE, A JAVASW INSTRUCTION SHOULD NOT BE USED IN THE LAST INSTRUCTION OF INSTRUCTION GROUP 255.

CONDITION CODE: INAFFECTED

INSTRUCTION FORMAT:

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0

INSTRUCTION FIELDS:

90

JAVA INTERPRETER SWITCH SUPPORT IS PROVIDED BY THE JAVASW INSTRUCTION. THIS INSTRUCTION CAUSES CONTROL FLOW TO BE DIRECTED INTO A TABLE OF INSTRUCTION GROUPS. MOST GROUPS CONSISTS OF 16 INSTRUCTIONS AND CORRESPOND TO A SINGLE JAVA BYTECODE.

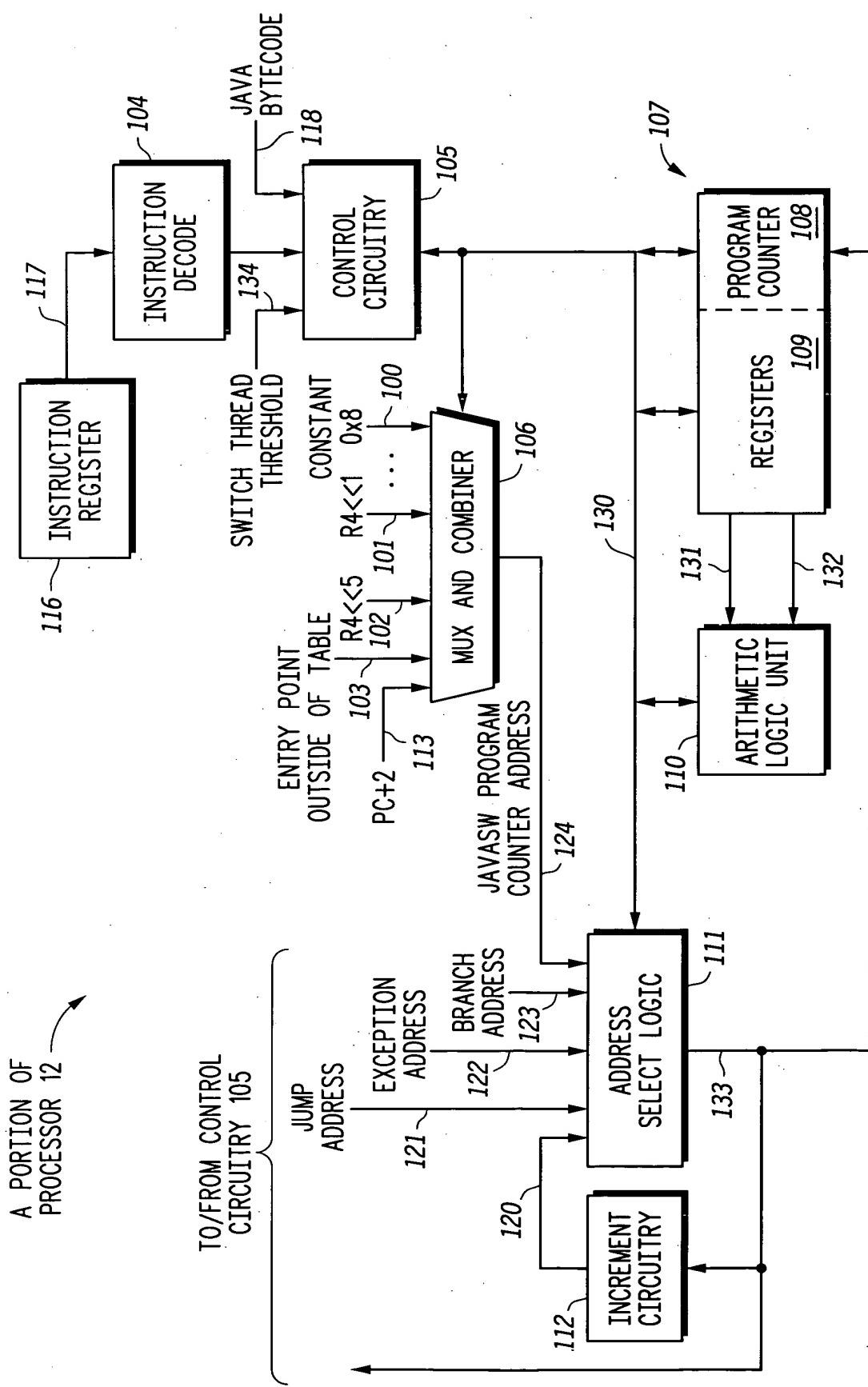


FIG. 8

7/7

EMULATION CODE
TABLE 83

4 INSTRUCTION SLOTS FOR THE "NOP" BYTECODE	84	16 INSTRUCTION SLOTS	87
12 INSTRUCTION SLOTS FOR THE THREADSWITCH ROUTINE	86		BYTECODE 0
			BYTECODE 1
			BYTECODE 2
			BYTECODE 3
			BYTECODE 4
		"IADD" ROUTINE	BYTECODE 0x60
		16 INSTRUCTION SLOTS	BYTECODE 0xbf
	88	1 INSTRUCTION SLOT	BYTECODE 0xc0
		1 INSTRUCTION SLOT	BYTECODE 0xc1
		1 INSTRUCTION SLOT	BYTECODE 0xfe
		1 INSTRUCTION SLOT	BYTECODE 0xff

FIG. 9